ME 391: Independent Study
Final Report
May 2020

Development of a Low-Cost Quadruped Robot

Gabrielle Conard and Professor Alexander Brown

While much progress has been made in the development of legged, and specifically quadruped, robots, these robots still face many challenges while navigating complex terrain. One feature that a quadruped robot must have is the ability to respond appropriately to unexpected obstacles, since an obstacle can disrupt the robot's balance or even damage the leg if it does not adjust its position accordingly. A way of approaching this issue is by sensing the excess force acting on the legs and creating active compliance in the joints of the robot. Active compliance describes the flexibility or softness of the joints in response to excess force from an impact. Such a feature is crucial for mitigating impacts that could damage the robot.

A number of methods have been used to create compliance. For example, the MIT Cheetah [1] and Ghost Robotics' Minitaur [2] use the motors themselves to detect and respond to impacts. Alternatively, other robots have used separate force sensors, typically installed in the feet, to sense excess force and adjust the motors appropriately [3, 4]. However, there is still much work to be done in reducing the cost of these machines. For example, the Minitaur is considered "relatively cheap", but costs about $10,000 [5]. The complexity and cost of them can be prohibitive to researchers [6]. As a result, in order to promote progress in this field, it is imperative to make quadruped technology more accessible through miniaturization and cost reduction, but that still have some of the abilities of their larger companions.

This semester, we set out to develop a low-cost quadruped robot capable of responding compliantly to its environment. Since inexpensive hobby servos are controlled using position commands rather than current adjustments as used in the MIT Cheetah and Ghost Robotics' Minitaur [1,2], the motors themselves cannot be used to sense and react to a force. One approach is to develop a separate force sensor and to model the joint's behavior as a virtual spring. After measuring the applied force, a spring constant, or a measure of the joint's desired stiffness, is used to calculate the distance the foot should retreat to compensate for the impact. While eventually this feature will be used while navigating an uneven terrain, the primary goal this semester was to build a robot that could respond compliantly to added weight or a small drop.

An Overview

The overall design was based heavily on the original MicroDog design, created by Professor Brown last semester as a proof-of-concept for the senior capstone project. It consists of a rectangular, two-layer chassis upon which the electrical boards and controllers are mounted. Attached to the bottom of the chassis is a tray to hold the battery. In total, the robot uses twelve motors to control the legs. As shown in Figure 1, the four hip servos, which control the frontal motion of the legs, are secured between the two chassis layers, while the femur (upper leg) and tibia (lower leg) servos, which control the sagittal and vertical motion, are mounted on the femurs.

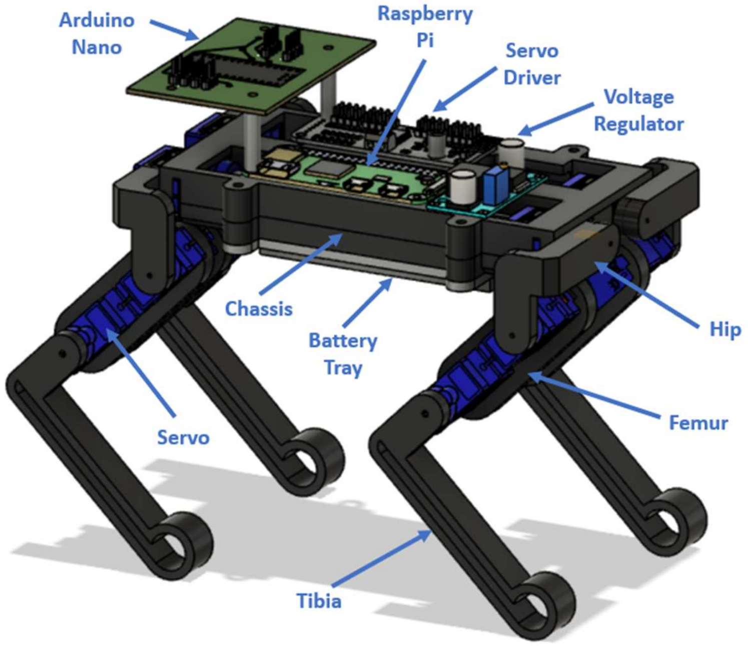

Figure 1: Original MicroDog design. Some modifications not shown include the force sensing tibias and updated chassis and battery tray.

The robot is controlled by a Raspberry Pi Zero W, which uses I^{2}C communication to interact with an Arduino Nano and a servo driver. This is all powered by a 7.4 V battery, though a voltage regulator adjusts the voltage across the Raspberry Pi to 5 V. In addition, we designed a four PCBs for the force sensors and a PCB to contain the Arduino Nano, I^{2}C buses, and force sensor connections, both of which will be discussed in more detail.

As presented in Table 1, the robot cost about $160, meeting our goal of developing an inexpensive robot.

Table 1: Cost Breakdown for modified MicroDog

Item	Cost/Pack	Items/pack	Unit Price	Quantity Used	Price of Used Components
Servo Driver	6.59	1.00	6.59	1.00	6.59
Battery	30.00	2.00	15.00	1.00	15.00
SD Card	8.49	1.00	8.49	1.00	8.49
Deans Connectors	6.99	10.00	0.70	1.00	0.70
MG90S Servos	29.95	10.00	3.00	12.00	35.94

Raspberry Pi Zero W	10.00	1.00	10.00	1.00	10.00
Voltage Regulator	5.99	2.00	3.00	1.00	3.00
2.5mm screw pack	12.00	1.00	12.00	1.00	12.00
Jumper Assortment	9.99	1.00	9.99	1.00	9.99
Arduino Nano	20.70	1.00	20.70	1.00	20.70
Arduino PCB	6.40	1.00	6.40	1.00	6.40
Force Sensor PCB	5.30	1.00	5.30	4.00	21.20
Hall Effect Sensor	1.95	1.00	1.95	4.00	7.80
6x3mm Magnets	12.98	100.00	0.13	4.00	0.52
Break Away Pin Headers	1.50	1.00	1.50	1.00	1.50
TOTAL					159.82

Sensing Forces: Design and Software

In order to create a virtual spring, it was necessary to develop a method of sensing the force being applied to the foot. We decided to design a flexible foot containing a hall effect sensor that would allow us to infer the magnitude of the contact force based on how much the foot bends, as shown in Figure 2. Set in a slot on top of each foot, each force sensor PCB contains a hall effect sensor and three jumpers that connect to an Arduino's analog pin. A magnet is inserted in the bottom of each foot. As force is applied, the top and bottom halves of the foot are squeezed together, bringing the hall effect sensor and the magnet closer together and increasing the magnitude of the sensor's signal. The stiffness of the foot can be adjusted by sliding the insert up and down.

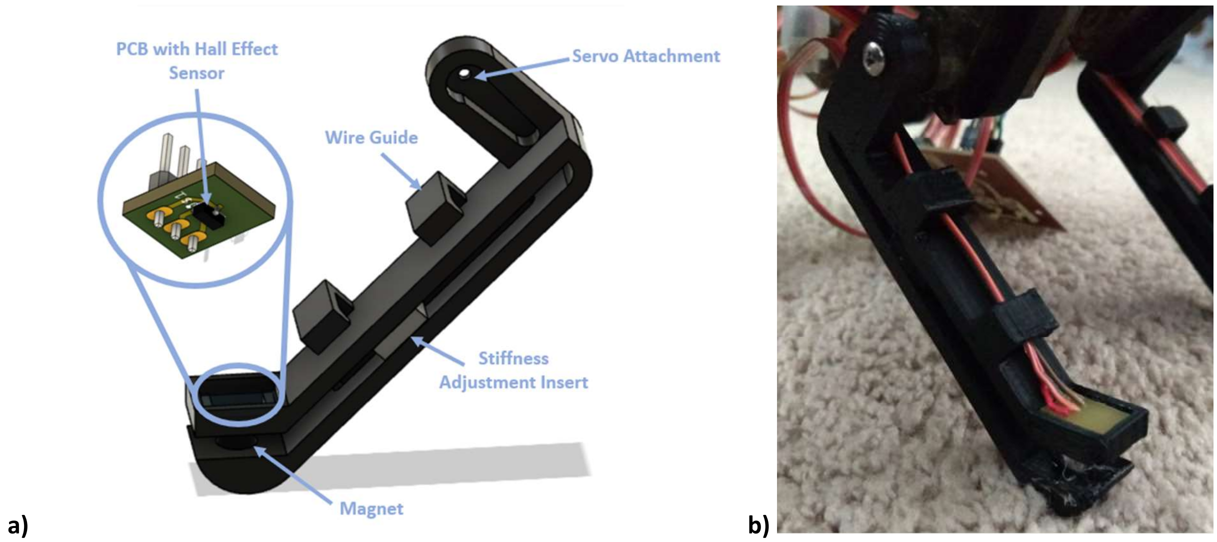

Figure 2: a) Model of tibia with force sensing foot. Inset is a closer look at the force sensor PCB, which contains the hall effect sensor. b) Force sensing tibia on the robot.

The Arduino Nano serves as the interpreter between the sensors and the Raspberry Pi, converting the analog signal to a digital signal. As a result, it was necessary to design a PCB that included signal, ground, and power pins for the sensors as well as two I^{2}C buses to communicate with the Raspberry Pi and servo driver. A closer look is provided in Figure 3.

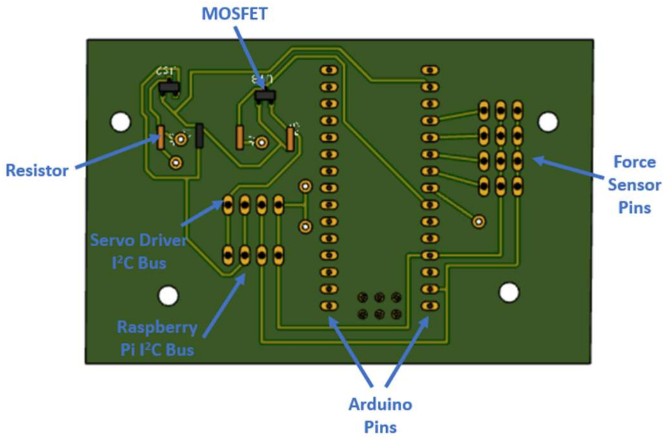

Figure 3: The Arduino Nano PCB consisted of ports for the force sensors and two I^{2}C buses.

Once the tibias and force sensors were designed and assembled, we performed several experiments to characterize the behavior of the flexible foot. First, we needed to calculate the conversion factor between signal received by the Raspberry Pi and the distance the foot compressed. Compressing the foot at small intervals, we measured the width of the gap between the top and bottom of the foot and recorded the corresponding sensor reading. After offsetting the sensor readings so that the full open gap corresponded with a reading of zero and converting the distance measurements to displacements, we fitted the data to a linear fit as seen in Figure 5.a, assuming that the slope would be the conversion factor.

To find the force from this distance measurement, we modeled the foot as a linear spring. Using a scale, we recorded the applied force and the corresponding sensor reading and distance, as seen in Figure 4. Then, since the spring constant is the slope of force over distance, we fitted the force and displacement data to a linear fit once again. See Figure 5.b for the resulting plot.

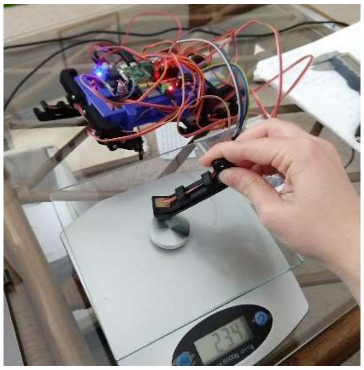

Figure 4: Measuring the applied force with a scale.

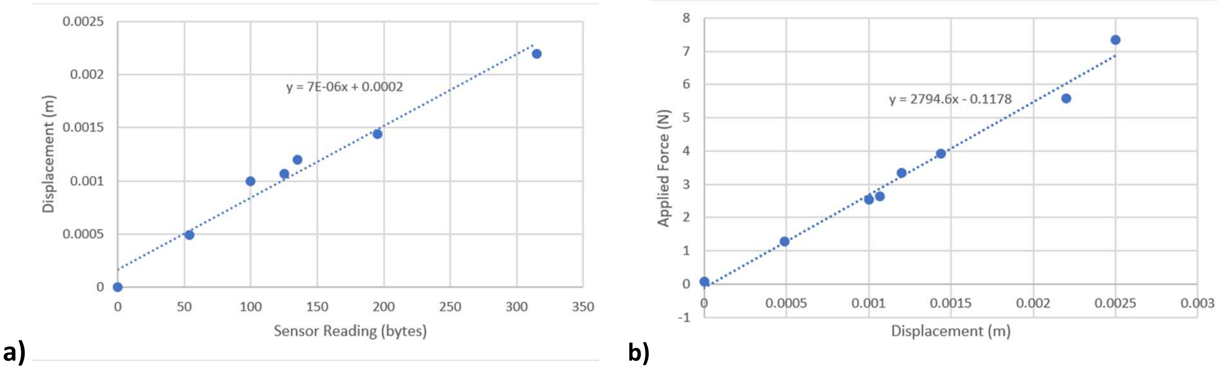

Figure 5: a) From the slope of the sensor data and measured displacements, we could determine the conversion factor. b) Similarly, the spring constant of the foot was found from measuring displacements and forces.

While we assumed that these relationships are linear, this assumption is not completely accurate, as evident in the plots in Figure 5. This is due to the fact that the foot is not a perfectly linear spring; instead, it appears that the foot becomes stiffer as the displacement increased, causing the conversion factor to decrease and the spring constant to increase. If this is the case, such an increase in stiffness would help prevent an unstable compensation response as the force increases. However, these experiments were by no means precise; for example, it was difficult to apply consistent force at the joint, leading to some error. Developing methods to make these measurements more consistently will help us to obtain a more accurate characterization of the foot. As it stands now, these results gave us good estimates to work with while developing the software.

Like the design itself, the software developed this semester was built on the code created for the MicroDog by Professor Brown, which included an inverse kinematics node, a library of movements and actions, scripts for communicating with the servo driver, and a central program to run them all. In addition to adding a node to establish I^2C communications with the Arduino Nano and separate the reading from each sensor into a separate message, several changes were made to the library SG90dog to enable active compliance, such as a function that takes in each sensor reading message and:

- Computes the corresponding foot displacement using the conversion factor found above
- Calculates the applied force using this displacement and the spring constant found above
- Determines the desired compensation using this force and the desired spring constant of the joints

We assumed that compensation was only necessary in the vertical direction, and as a result the vertical position of the foot determined by the robot's current action (such as standing or walking) is offset by the compensation factor. This is then passed to the inverse kinematics node, which computes the appropriate servo angles to achieve that position.

Testing

As a location for a payload has not yet been incorporated in the design, added weight was simulated by pressing down on the chassis or squeezing a foot. By making the virtual spring constant much smaller than the foot spring constant, the leg was able to fold in response to the increased force, as shown in Figure 6.

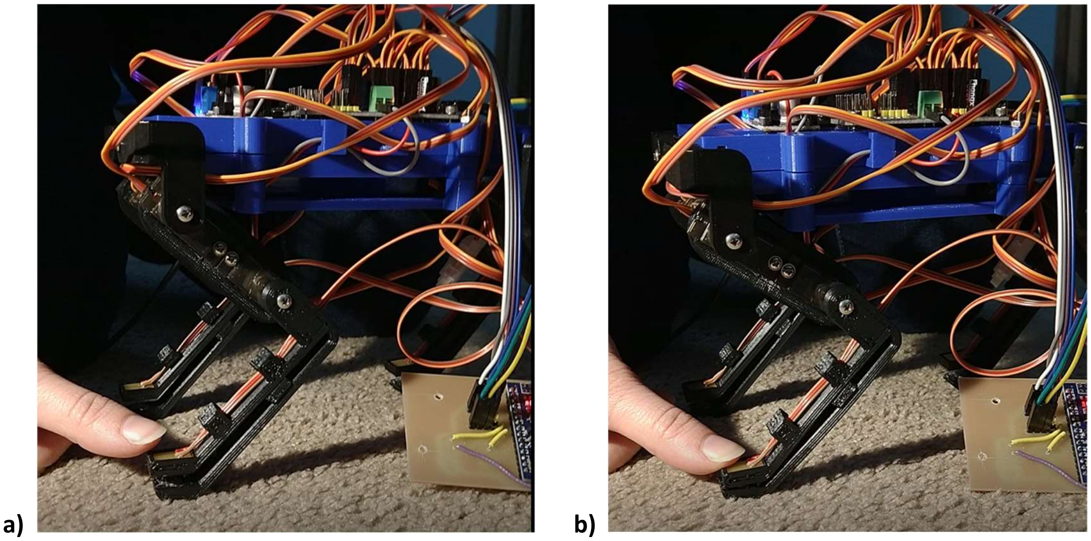

Figure 6: a) Left foreleg shown in normal stand position. b) Leg bends due to compression of foot.

Pressing directly on the foot, however, is not a very accurate approximation of added weight, since doing so forces the foot to remain stationary. When weight is applied to a corner, the compensation stops the foot from being compressed so far but also allows the foot to slide forward and out from underneath the hip, further decreasing the compression of the foot and the compensation. This is problematic as the leg is not compensating predictably. In addition, the rear feet tend to slide too far forward for the robot to push off the ground and stand up again after weight is removed.

The robot was also subjected to several drop tests. It appeared that the rear feet tended to compress the most, causing the rear legs to compensate most as well. This may be due to the robot's center of gravity being shifted slightly towards the rear. However, this causes uneven compensation and sometimes causes the robot to fall over. While the robot did not fall during the test shown in Figure 7, this leaning is apparent in 7.3. In addition, due to the redesigned battery tray, the front legs cannot fold all the way without colliding with the tray. This prevents the front legs from being able to compensate as much as the rear. As a result, a possible solution is to redesign the battery tray so that the front legs have sufficient room to fold. In addition, it may be helpful to extend the legs from their normal standing positions to allow more room to fold and soften the impact.

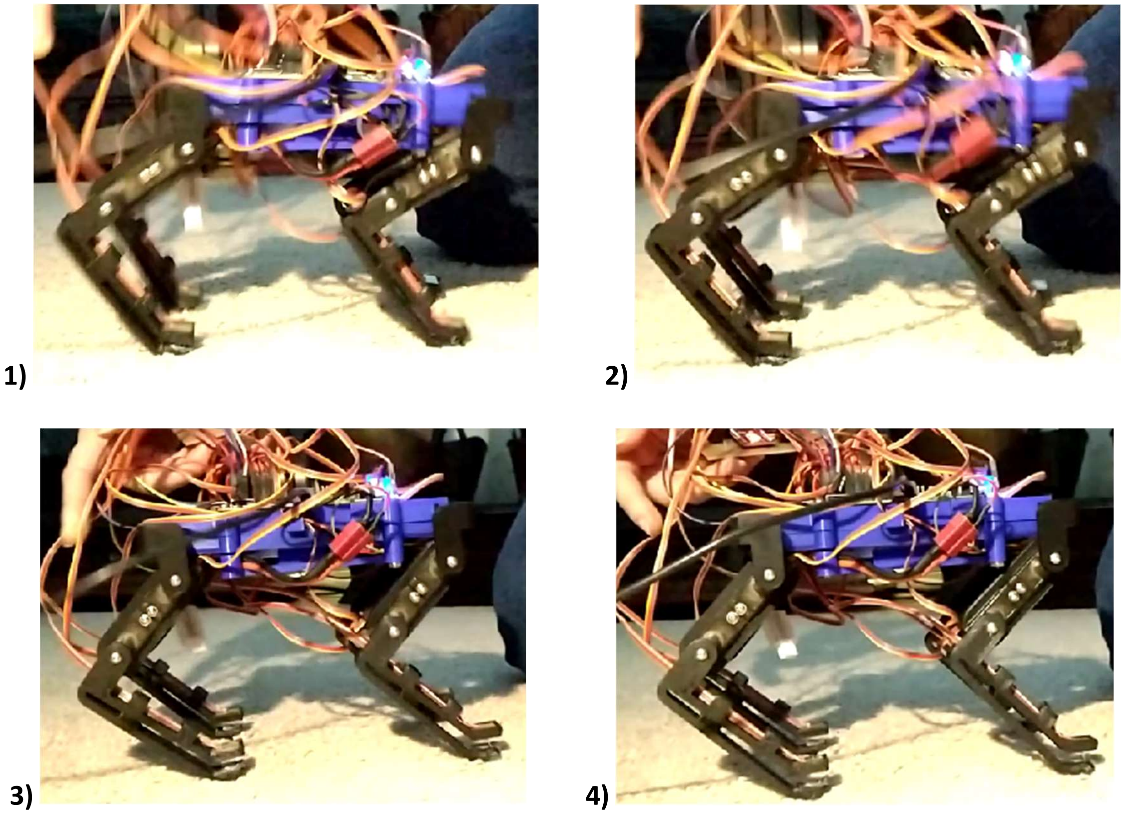

Figure 7: A series of images from a drop test. 1) Compression of front feet is evident, and leg has bent to compensate for the increased force. 2) Robot begins to bounce backwards, leading to more even compression of all feet and some compliance in all legs. 3) Robot leans back on rear legs, allowing the front feet to nearly lift off the ground. 4) At the end, the robot finally settles into a more balanced position, though it appears that the rear legs did not completely return to their standing positions.

Improvements and Further Research

While some advances were made this semester, there are many ways this robot can be improved. In terms of the physical and electrical design, we would suggest the following:

- Redesign the battery tray and the bottom of the chassis. In the current design, the front legs cannot fold completely without colliding with the battery tray, which limits the range of motion. It may also be helpful to limit the maximum servo angles in the software.
- Improve the stability of the robot. As of now, the robot is somewhat top-heavy, likely due to its long, narrow design. The instability is especially noticeable when it is walking. Adding an accelerometer may also help increase the robot's balance.
- Develop one PCB to contain all of the boards. This would improve organization on the chassis and hopefully solve some connection issues.
    - For example, one of the Arduino signal pins is receiving an occasional spike whenever one of the other pins is receiving an increased signal. This leads to violent motions in the leg controlled by that pin. If a new PCB does not fix the issue, potentially a median filter could be used to remove spikes in the signal.
- Create a method for calibrating the force sensors at start-up, since their base readings tend to vary slightly by the day.

In addition, there are several possible extensions to this research. One path is to investigate several other force sensor designs. For example, Rachel Sloan developed a 3D printed strain gage that can be used to measure force. Printing such a sensor in the shape of a tibia would allow use to turn the entire lower leg into a force sensor. The strain gage design can be characterized and compared to the hall effect sensor to determine which is more effective. Another extension is to investigate more deeply how quadruped animals absorb impacts and added weight. One issue that we noticed was that when weight was added to the rear of the robot, the rear feet tended to slide forward even on a rougher surface. Since the feet were no longer aligned underneath the hips, the legs were unable to extend and return to their standard positions when weight was removed. A possible cause is that we are only accounting for vertical compensation. In reality, it is possible that animals lean forward or backwards in the process of standing up or lying down. Further studying the movement of animals would help us determine if it is necessary to compensate in the horizontal direction as well.

Finally, active compliance can also be incorporated into walking to detect obstacles. With this added feature, it would be useful to develop different compliance modes that use different spring constants. For example, a stiffer spring constant can be used while the robot is walking to prevent excess jittering. When the robot detects that more than two feet are off the ground, it can enter a drop test mode and uses a softer spring constant to absorb the impact from a drop. Developing modes such as this would allow the robot to respond more efficiently to the complex world around it.

References

[1] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo and S. Kim, "Optimized Jumping on the MIT Cheetah 3 Robot," *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 7448-7454.

[2] G. Kenneally, A. De and D. E. Koditschek, "Design Principles for a Family of Direct-Drive Legged Robots," in *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900-907, July 2016.

[3] X. Li, W. Wang and J. Yi, "Foot contact force of walk gait for a quadruped robot," *2016 IEEE International Conference on Mechatronics and Automation*, Harbin, 2016, pp. 659-664.

[4] J. Xu, L. Lang, H. Ma and Q. Wei, "Contact force based compliance control for a trotting quadruped robot," *The 27th Chinese Control and Decision Conference (2015 CCDC)*, Qingdao, 2015, pp. 5144-5149.

[5] E. Ackerman, “Ghost Robotics' Minitaur Demonstrates Impressive New Skills,” IEEE Spectrum, https://spectrum.ieee.org/automaton/robotics/robotics-hardware/ghost-robotics-minitaur-demonstrates-impressive-new-skills (accessed May 19, 2020).

[6] K. G. Karwa, S. Mondal, A. Kumar and A. Thakur, "An open source low-cost alligator-inspired robotic research platform," *2016 Sixth International Symposium on Embedded Computing and System Design (ISED)*, Patna, 2016, pp. 234-238.